REMARKS

Claims 1-50 are pending in the present application. By this Response, claims 1, 19 and 35 are amended to correct antecedent basis problems.

No new matter is added by the above amendments. Reconsideration of the claims in view of the above amendments and the following Remarks is respectfully requested.

I. 35 U.S.C. § 112, Claim 1

The Office Action rejects claim 1, which recites, "associatively storing an identifier for each of said attributes" in lines 10-11 on page 25 of the current specification, under 35 U.S.C. § 112, because there is insufficient antecedent basis for this limitation in the claim. By this Response, claim 1 is amended to recite, "associatively storing an identifier for each of the user-defined attributes and each of the PKCS-standard defined attributes". Claims 19 and 35 are also amended to be consistent with claim 1. Therefore, claims 1, 19, and 35 now have sufficient antecedent basis and are now in condition for allowance. Accordingly, Applicants respectfully request the withdrawal of the rejection to claim 1 under 35 U.S.C. § 112.

II. 35 U.S.C. § 103(a), Obviousness, Claims 1-50

The Office Action rejects claims 1-50 under 35 U.S.C. § 103(a) as being unpatentable over Katin et al. (U.S. Patent No. 5,261,098) in view of RSA (PKCS#9 v2.0). This rejection is respectfully traversed.

As to claim 1, the Office Action states:

> With respect to Claim 1, the limitation "associatively storing an identifier for each of said attributes" is met by Katin on column 6, lines 67-68 and on column 7, lines 1-2.
>
> Further limitation of "registering attributes..." is inherently met by Katin on column 1, lines 24-37. In this referenced section, it is inherent in the object-oriented programming method that the object be registered with a class or an object type family. This is also discussed briefly on column 9, lines 1-7 of Katin.

> Katin however does not disclose registering these attributes with a PCKS9 gateway class. This is however disclosed by RSA as shown below.
>
> The limitation "registering attributes with a PCKS9 gateway class, wherein the attributes include user-defined attributes and PKCS-standard (Public Key Cryptography Standards) defined attributes" is met by RSA on page 5, section 4.1.
>
> It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of RSA within the teachings of Katin et al because the method steps described are unique to a system that would want to register and store an identifier for the purpose of utilizing object oriented methods in efficiently storing and classifying data. Katin et al claims on column 2, lines 5-10 that these object-oriented methods "improve inter-operability between applications..." since "it is often desirable for the applications to be able to invoke each other and/or share objects." This would prove useful in implementing these object-oriented methods within a PKCS9 gateway class because this class 'defines a set of attributes that can be used in other PKCS standards'. Hence achieving optimal interoperability is the goal of the PKCS9 gateway class.
>
> Hence, it would have been obvious to combine the RSW teaching within the system of Katin et al to achieve the claimed invention.

Office Action dated January 29, 2004, pages 1-2.

Amended independent claim 1, which is representative of independent claims 19 and 35 with regard to similarly recited subject matter, now recites:

> 1. A method in a data processing system for managing data attributes, the method comprising the steps of:
>
> registering attributes with a PKCS9 gateway class, wherein the attributes include user-defined attributes and PKCS-standard (Public Key Cryptography Standards) defined attributes; and
>
> associatively storing an identifier for each of the user-defined attributes and each of the PKCS-standard defined attributes.(emphasis added)

Katin teaches a method and apparatus for deriving an object's object type and obtaining object type attribute values for the derived object type for an application:

> An object type and at least one object type deriving attribute are stored as an object type entry in an object type table having an object type table identifier. The object type table is in turn stored in a database. Similarly, the object type and at least one object type attribute value having a corresponding object type attribute identifier are stored as an object type attribute entry in an object type attribute table having an object type attribute table identifier. The object type attribute table is also stored in the database.

> An object type deriving manager and an object type attribute values obtaining manager corresponding to the object type table and the object type attribute table are provided for deriving object type and obtaining object type attribute values respectively. A row getting interface routine and a column getting interface routine are provided for invoking the functions of the object type deriving manager and the object type attribute values obtaining manager.
>
> An object's object type and the object type's attribute values are derived and obtained by providing an object type table identifier, at least one object type deriving attribute, and an object type attribute table identifier, and at least one object type attribute identifier to the row and column getting interface routines. In response, the row and column getting interface routines derive the object's object type from the stored object type entry, obtain the derived object type's object type attribute values from a stored object type attribute entry and returned the derived object type and the obtained object type attribute values to the application using the functions of the object type deriving and the object type attribute values obtaining managers.

(Column 2, line 36 to column 3, line 12, Katin)

RSA teaches a proposed draft #3 of the PKCS #9 version 2.0 standard, which includes two new auxiliary object classes, pcksEntity and naturalPerson, as well as selected attribute types for use with these classes. Furthermore, it defines attribute types for use in conjunction with PKCS #7 digitally signed messages, PKCS #10 certificate-signing requests, PKCS #12 personal information exchanges and PKCS #15 cryptographic tokens and matching rules for use with these attributes.

Neither Katin nor RSA teach or suggest registering attributes with a PKCS9 gateway class, wherein the attributes include user-defined attributes and PKCS-standard defined attributes. The Office Action alleges that these features are inherently met by Katin, since it is inherent in the object-oriented programming method, as taught by Katin at column 1, lines 24-37 and column 9, lines 1-7, that the object is registered with a class or an object type family. Applicants respectfully disagree. At column 1, lines 24-37, Katin teaches that an object in object-oriented programming includes data and operations which can be invoked to manipulate the data. An object has an object type, which defines object operations that can be performed on objects of that particular object type. At column 9, lines 1-7, Katin teaches that the object type attribute values obtaining manager obtains an object type attribute entry identifier or object type attribute values, and the functions provide the obtaining manager are dependent on the object type family and the format of the object type attribute table used.

While Katin teaches that data (attributes) and operations may be registered with an object or object type based on the object type family, it is not inherent in Katin that the attributes registered with the object or object type includes PKCS-standard defined attributes. According to the MPEP section 2112, Requirements of Rejection Based on Inherency; Burden or Proof, it states,

> "To establish inherency, the extrinsic evidence must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient."

In the referenced sections above, Katin only teaches that attributes or methods may be registered with an object in object-oriented programming for an application. However, PKCS-standard defined attributes are not necessarily present in the attributes as taught by Katin. A person of ordinary skill in the art would recognize that it is not necessary in object-oriented programming for attributes or methods in an object to include a PKCS-standard defined attributes. Since there is no teaching or suggestion in Katin to include PKCS-standard defined attributes, a person of ordinary skill in the art would not have included PKCS-standard defined attributes in an object in object-oriented programming without the disclosure of Applicants. Although it is possible that PKCS-standard defined attributes may be included in attributes of an object in object-oriented programming, inherency may not be established by the possibility. Therefore, Applicants respectfully submit that it is not inherent in Katin that attributes registered with PKCS9 gateway class includes PKCS-standard defined attributes.

In addition, the Office Action admits that Katin does not disclose registering these attributes with a PKCS9 gateway class, but the Office Action alleges that RSA teaches these features on page 5, section 4.1. Applicants respectfully disagree. In section 4.1, RSA teaches a pkcsEntity object class that is intended to hold attributes about PKCS-related entities. Below is a definition of the pkcsEntity class:

```
pkcsEntity OBJECT-CLASS ::=     {
        SUBCLASS OF {top}
        KIND auxiliary
        MAY CONTAIN {PKCS9AttributeSet}
        ID pkcs-9-pkcsEntity
```

```
}

PKCS9AttributeSet ATTRIBUTE ::=        {
        userPKCS12 |
        pKCS15Token |
        encryptedPrivateKeyInfo,
        ...For future extensions
}
```

In the above definition, pkcsEntity class includes a PKCS9AttributeSet, which includes three attribute types: userPKCS12, pKCS15Token, and encryptedPrivateKeyInfo. Section 5.1 further describes these three attribute types. The userPKCS12 attribute represents PKSC #12 token that provides a format for exchange of personal identity information. The pKCS15Token attribute represents PKSC #15 token that provides a format for cryptographic tokens. The encryptedPrivateKeyInfo attribute represents PKSC #8 which provides a format for encrypted private keys.

However, none of the three attribute types mentioned above in the RSA reference includes user-defined attributes. To the contrary, the RSA reference as a whole teaches only the PKCS #9-standard defined attributes. On page 35 of the RSA reference, RSA teaches that the Public Key Cryptography Standards are specifications produced by RSA Laboratories in cooperation with secure systems developers worldwide for the purpose of accelerating the deployment of public-key cryptography. On page 33 of the RSA reference, a revision history shows that the RSA reference incorporates changes to version 2.0 draft 2 and other previous versions of the PKCS #9 standard. Thus, the RSA reference itself is a document that defines the standards. There is simply no suggestion of attributes other than PKCS standard attributes in the RSA reference.

In addition, in sections 5.2 and 5.3, RSA teaches a naturalPerson object class that holds attributes about human beings. These attributes include EmailAddress, UnstructuredName, UnstructuredAddress, MessageDigest, SigningTime, CounterSignature, ChallengePassword, ExtendedCertificateAttributes, and ContentType. All of these attributes are described in Figure 6 and on page 15, lines 7-17 of the current specification as attributes that belongs to a PKCS #9-standard defined attributes. User-defined attributes, as described on page 15, lines 18-24 of the current specification, are not defined in the PKCS #9 standard.

Therefore, the pkcsEntity object class as taught by RSA is not the same as the PKCS9 gateway class as recited in claim 1. Although the pkcsEntity object class in RSA includes PKCS standard attributes, it does not include user-defined attributes. Other object classes taught by RSA, such as the naturalPerson object class, also do not include user-defined attributes. They only include attributes that are PKCS-standard defined. Furthermore, the RSA reference only describes the PKCS #9 standard as a whole-it does not disclose user-defined attributes that are not within the standard. Hence, RSA does not teach registering attributes with a PKCS9 gateway class, wherein the attributes includes PKCS-standard defined attributes and user-defined attributes, as recited in claim 1. Since RSA is only concerned with PKCS-standard defined attributes, a person of ordinary skill in the art would not have been motivated to take RSA's pkcsEntity object class and register user-defined attributes with it.

The Office Action further alleges that it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of RSA within the teachings of Katin et al. because the method steps described are unique to a system that would want to register and store an identifier for the purpose of utilizing object-oriented methods in efficiently storing and classifying data. The Office Action states that column 2, lines 5-10 of Katin that these object-oriented methods "improve inter-operability between applications..." since "it is often desirable for the applications to be able to invoke each other and/or share objects." The Office Action further states that this would prove useful in implementing these object-oriented methods within a PKCS9 gateway class because this class 'defines a set of attributes that can be used in other PKCS standards'. The Office Action also states that achieving optimal interoperability is the goal of the PKCS9 gateway class.

At column 2, lines 5-10, Katin suggests that to further improve interoperability between applications, it is often desirable for application to be able to invoke each other and/or share objects. The various applications must be able to identify the objects' object types, their methods, icons, and other object type attributes, etc. Thus, Katin teaches a system that allows different object types to be quickly identified and different object type attribute values to be quickly obtained by different applications, such that interoperability between applications can be improved. However, Katin does not teach or suggest that the

different attributes include PKCS standard defined attributes or registering the different attributes with a PKCS9 gateway class. Nowhere in the reference does Katin mention anything about PKCS9 standard, let alone a PKCS9 gateway class that defines a set of attributes to be used with other PKCS standards, as alleged in the Office Action.

RSA also does not teach or suggest the alleged combination or the motivation for the alleged combination. RSA only teaches a PKCS #9 standard that includes a pkcsEntity object class and a naturalPerson object class. While these two object classes include attribute types of other PKCS standards, nowhere in the reference does RSA teach or suggest that the two object classes include user-defined attributes, which are not within the PKCS #9 standard. Therefore, RSA only suggests interoperability of attributes between applications with PKCS standards. RSA does not suggest interoperability between PKCS standard defined applications and user-defined (non-PKCS standard defined) applications.

Hence, it would not have been obvious to a person of ordinary skill in the art to combine the RSA teaching with the system of Katin to achieve the presently claimed invention, because there is no teaching or suggestion in either Katin or RSA to register user-defined attributes and PKCS standard defined attributes with a PKCS9 gateway class in order to improve interoperability between PKCS related applications and user-defined applications. Even if a person of ordinary skill in the art is to combine the teachings of Katin and RSA, the result would not be the same as the presently claimed invention. The result would be registering PKCS standard defined attributes and methods with a pkcsEntity object class. The result would not be registering PKCS standard defined and user defined attributes with a PKCS9 gateway class, as recited in claim 1.

In view of the above, Applicants respectfully submit that neither Katin nor RSA teach or suggest all of the features of independent claim 1. The other independent claims 19 and 35 recite similar features also not taught or suggested by either reference. Accordingly, Applicants respectfully request the withdrawal of the rejection of claims 1, 19 and 35 under 35 U.S.C. § 103(a). At least by virtue of their dependency on claims 1, 19 and 35, respectively, neither Katin nor RSA teach or suggest the features of claims 2-3, 20-21 and 36-37. Accordingly, Applicant respectfully requests withdrawal of the rejection of claim 1-3, 19-21 and 35-37 under 35 U.S.C. § 103(a).

In addition, neither Katin nor RSA teach or suggest the specific features recited in dependent claims 2-3, 20-21 and 36-37. For example, with regard to claim 2, which is representative of claims 20 and 36 with regard to similarly recited subject matter, neither Katin nor RSA teach or suggest calling a first object-oriented method in the PKCS9 gateway class, wherein the first object-oriented method receives a parameter comprising an object identifier for an attribute. The Office Action alleges that Katin teaches these features at column 9, lines 36-38, which reads as follows:

> The object type deriving matching routine 57 receives an object type entry identifier and at least one object type deriving attribute as input arguments.

As described above, Katin does not teach a PKCS9 gateway class that includes a user-defined attributes and PKCS-standard defined attributes. In addition, in the above section, Katin teaches that the matching routine receives as its input arguments an object type entry identifier that identifies an entry in the object type table and one object type deriving attribute. **Figure 4a** of Katin, which describes the object type table, is shown below:

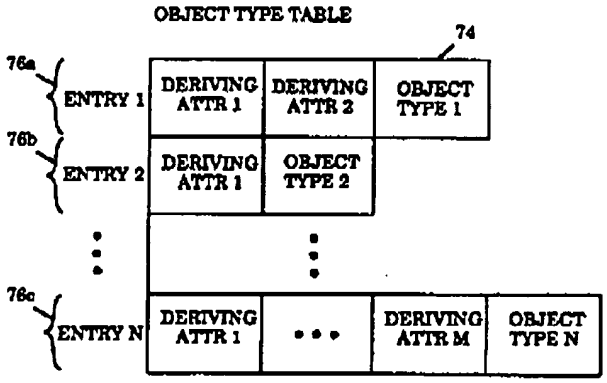

Figure 4a

As shown in **Figure 4a**, the object type table includes a plurality of object type entries. Each object type entry **76a, 76b, 76c** comprises a plurality of object type

deriving attributes and an object type. Each object type entry **76a, 76b, 76c** is identified by an object type entry identifier.

Thus, Katin only teaches a matching routine (method) that receives object type entry identifiers, such as Entry 1 **76a**, and a deriving attribute, such as Deriving Attr 1, as input parameters. Object type entry identifier identifies an entry in the object type table and the deriving attribute identifies an attribute for an object type. Neither of these two input parameters identifies an object identifier for an attribute. Therefore, Katin does not teach a first object-oriented method in a PKCS9 gateway class that receives a parameter comprising an object identifier for an attribute, as recited in claims 2, 20 and 36.

RSA also does not teach calling a first object-oriented method in the PKCS9 gateway class, wherein the first object-oriented method receives a parameter comprising an object identifier for an attribute. RSA only teaches a standard of PKCS defined attributes. While RSA teaches an object identifier that is defined in an attribute, RSA does not teach any method that receives a parameter comprising the object identifier for an attribute. This is because RSA only teaches attributes selected for the PKCS #9 standard, RSA does not teach any implementation detail of such standard.

With regard to dependent claim 3, which is representative of claims 21 and 37, neither Katin nor RSA teach or suggest searching an attribute mapping data structure using the object identifier in the received parameter, in response to a determination of a matching object identifier in the attribute mapping data structure, retrieving a class identifier associatively stored with the matching object identifier in the attribute mapping data structure, or calling a second object-oriented method in a class identified by the retrieved class identifier. The Office Action alleges that Katin teaches these features at column 9, lines 38-56, which reads as follows:

> In response, if the deriving attribute input arguments match the object type deriving attributes in the object type entry identified by the object type entry identifier, the object type deriving matching routine **57** returns the object type entry identifier, else it returns a null value.
>
> The object type retrieval chaining key routine **59** receives an object type entry identifier as input argument. In response, the object type retrieval chaining key routine **59** returns a chaining key. The chaining key comprises the object type deriving attributes. The object type retrieval chained entry matching routine **61** receives an object type entry identifier and a chaining key as input arguments. In response, if the chaining key input argument intersects with a subset of the

> object type deriving attributes of the object type entry identified by the object type entry identifier input argument, the object type retrieval chained entry matching routine returns a TRUE value, else it returns a FALSE value. (emphasis added)

In the above section, Katin teaches three routines: the object type deriving matching routine, the object type retrieval chaining key routine and the object type retrieval chained entry matching routine. The object type deriving matching routine takes deriving attributes as input arguments and returns an object type entry identifier if input deriving attributes match the deriving attributes identified by the object type entry identifier or a null value. The object type retrieval chaining key routine takes an object type entry identifier as input argument and returns a chaining key. The object type retrieval chained key entry matching routine takes an object type entry identifier and a chaining key, which comprises object type deriving attributes, as input arguments. The matching routine returns a TRUE value if the deriving attributes of the input chaining key intersects a subset of the object type deriving attributes of the object type entry identified by the input object type entry identifier. Otherwise, it returns a FALSE value.

Thus, Katin only teaches searching an attribute mapping data structure (object type table) using an object type entry identifier, a chaining key, or deriving attributes as its input arguments. As described above, the object type entry identifier identifies an entry in the object type table, the deriving attributes identify attributes of an object type and the chaining key also identifies a set of deriving attributes. None of these input arguments identify an object identifier for an attribute. Therefore, Katin does not teach searching an attribute mapping data structure using an object identifier in the received parameter.

In addition, nowhere in the above section, or any other section, of the reference does Katin teach or suggest retrieving a class identifier associatively stored with the matching object identifier or calling a second object-oriented method in the class identified by the retrieved class identifier. Katin only teaches retrieving an object type entry identifier, a chaining key or a TRUE or FALSE value. None of the return values includes a class identifier. Katin also fails to mention anything about a class identifier that corresponds to an object identifier in the object type table, Katin only teaches an object type that corresponds to a plurality of object type deriving attributes in the object

type table. Katin does not teach retrieving a class identifier. Since Katin does not teach anything about retrieving a class identifier, Katin does not and would not teach calling a method of the class identified by the retrieved class identifier. Therefore, Katin does not teach or suggest the features of claims 3, 21 and 37.

RSA also does not teach the features of claims 3, 21 and 37. As described above, RSA only teaches a standard of PKCS defined attributes. While RSA teaches an object identifier that is defined in an attribute, RSA does not teach searching the mapping data structure, matching the object identifier in the data structure or retrieving the identifier from the attribute in the attribute mapping data structure since RSA does not teach any implementation detail.

In view of the above, Applicants respectfully submits that neither Katin nor RSA teach or suggest the specific features of dependent claims 2-3, 20-21 and 36-37. Thus, Applicants respectfully request withdrawal of the rejection of claims 2-3, 20-21 and 36-37 under 35 U.S.C. § 103(a).

Independent claim 4 recites:

> 4. A method in a data processing system for managing data attributes, the method comprising the steps of:
>
> invoking a first object-oriented method to process an attribute object, wherein the first object-oriented method is defined in an abstract class for attribute objects with a subclass for undefined attributes and a subclass for defined attributes, wherein the subclass for defined attributes is further comprised of a subclass for each PKCS-defined (Public Key Cryptography Standards) attribute and a subclass for each user-defined attribute;
>
> invoking a second object-oriented method to process an attribute object, wherein the second object-oriented method is defined in a PKCS9 gateway class; and
>
> in response to invoking the first object-oriented method or the second object-oriented method, processing the result returned by the first object-oriented method or the second object-oriented method. (emphasis added)

Neither Katin nor RSA teaches the features emphasized above. The Office Action alleges that Katin teaches these features at column 9, lines 36-38 and 51-56, which is reproduced above, and at column 9, lines 1-14 and column 3, lines 13-20, which reads as follows:

> Similarly, the manner in which an object type attribute values obtaining manager **58a** or **58b** obtains an object type attribute entry identifier or object type

attribute values, and therefore the functions provided by an object type attribute values obtaining manager **58a** or **58b**, are dependent on the object type family and the format of the object type attribute table used. For the exemplary object type attribute table format discussed earlier (**FIG. 4b**), the object type attribute values obtaining manager **58a** or **58b** may obtain the object type attribute entry identifier by matching the input arguments against the object type in the object type attribute entries and retrieving the object type attribute values from the matched object type attribute entry.

Under the present invention, different object types may have different deriving object type attributes and different object type attributes. The object type and/or object attribute tables may be locally and/or regionally customized. The customized versions of an object type attribute table as well as the customized versions of an object type attribute table are stored in different databases. (emphasis added)

Figure 4b of Katin, which describes the object type attribute table, is shown below:

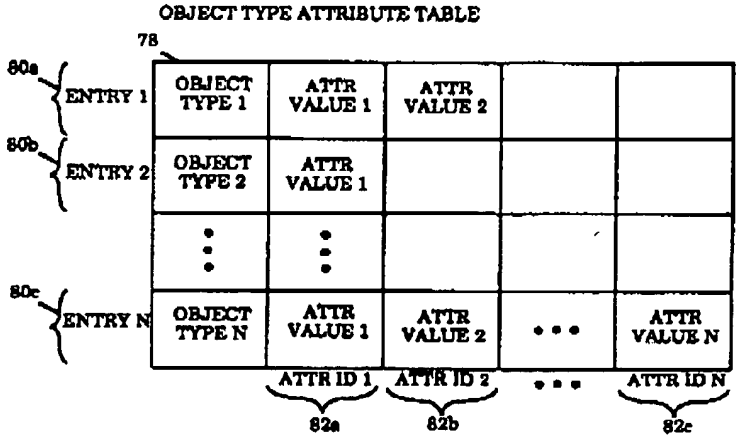

Figure 4b

As shown in **Figure 4b**, the object type attribute table **78** comprises a plurality of object type attribute entries **80a-80c**, one for each object type in the object type family. Each object type attribute entry **80a-80c** comprises an object type and a plurality of object type attribute values. Each object type attribute entry **80a-80c** is identified by an object type attribute entry identifier, and each object type attribute value is identified by an object type attribute identifier **82a-82c**.

In the above sections, Katin only teaches an object type attribute values obtaining manager that obtains an object type attribute entry identifier or object type attribute

values if a matching object type or object type attribute entry is found. However, Katin does not teach a routine that is defined in an abstract class that represents attribute objects. Although Katin teaches routines that interact with the customized object type table and object type attribute table, these routines are defined in the object type attribute values obtaining manager and are separated from the attributes in the tables. Nowhere in the reference docs Katin teach or suggest a routine that is defined in an abstract class for attribute objects. RSA also does not teach any routine defined in the PKCS #9 standard, only attributes.

In addition, Katin does not teach an abstract class for attribute objects with a subclass for undefined attributes and a subclass for defined attributes, as recited in claim 4. Katin only teaches at column 7, lines 47-64 that customized object type attribute values may be stored in the customized version of object type attribute table. Thus, Katin allows different object type attribute values to be scattered among multiple customized versions of the object type attribute table. Examples of object type attribute values include icon file name, foreground colors, background colors, icon mask name, etc. However, Katin does not mention anything about attributes that are undefined being stored in the object type attributes table.

As described on pages 15-16 of the current specification, undefined attributes represent attributes that are unknown to the system, which are neither PKCS standard defined nor user-defined. Katin fails to teach storing an object type attribute that is unknown to an application. Katin only teaches customized versions of the object type attribute table, which include customized object type attribute values. Therefore, Katin does not teach an abstract class with a subclass for undefined attributes, as recited in claim 4. RSA also does not teach any abstract class with a subclass for undefined attributes. As discussed in the arguments presented for claim 1, RSA only teaches PKCS standard defined attributes, RSA does not teach any undefined attributes.

Furthermore, neither Katin nor RSA teach or suggest the subclass for defined attributes is further comprised of a subclass for each PKCS-defined (Public Key Cryptography Standards) attribute and a subclass for each user-defined attribute. The Office Action admits that Katin does not teach a PKCS-defined attribute, but the Office

Action alleges that RSA teaches this feature in the last two paragraphs on page 8, section 5.2.2, which reads as follows:

> The PKCS9String type is defined as a choice of IA5String and DirectoryString. Applications SHOULD use the IA5String type when generating attribute values in accordance with this version of this document, unless internationalization issues makes this impossible. In that case, the UTF8String alternative of the DirectoryString alternative is the preferred choice. PKCS#9-attribute processing systems MUST be able to recognize and process all string types in PKCS9String values.
>
> Note. Version 1.2 of this document defined unstructuredName as having the syntax IA5String, but did contain a note explaining that this might be changed to a CHOICE of different string types in future versions. To better accommodate international names, this type has been extended to also include a directory string in this version of this document. Since [21] does not support a directory string type containing IA5Strings, a separate syntax object identifier has been defined (see [21] and Appendix B).

In the above sections, RSA teaches within the attribute type UnstructuredName, which specifies name or names of a subject as an unstructured ASCII string, a PKCS9String attribute type is defined. The PKCS9String type includes a choice of IA5String type or DirectoryString type. RSA teaches that applications should use the IA5String type unless internationalization becomes an issue. Thus, RSA merely teaches a PKCS9String type that is defined within an UnstructuredName attribute type for representing names of a subject. As described above for claim 1, the RSA reference as a whole discloses what attribute types are to be selected for the PKCS #9 standard. However, the RSA reference does not teach or suggest anything about a subclass for each user-defined attribute. As described above, RSA is only concerned with PKCS standard defined attributes, not user-defined attributes that are not part of the standard. Therefore, RSA does not and would not teach a subclass for each user-defined attributes.

A person of ordinary skill in the art would not have been motivated to modify Katin's object type attributes table to include RSA's PKCS defined attributes to arrive at the presently claimed invention, because neither Katin nor RSA teach or suggest a method that is defined in a subclass for attribute objects with a subclass of defined attributes and a subclass of undefined attributes. Katin only teaches defining a routine in an object type attribute values obtaining manager that obtains attribute values from an object type attribute table. Katin does not teach an abstract class that represents attribute

objects. Katin also does not teach any subclass for undefined attributes. RSA also does not teach these features. While RSA teaches PKCS defined attributes, there is no mention of any abstract class for attributes objects that has a subclass of undefined attributes. In addition, RSA is only concerned with PKCS standard defined attributes, RSA is not concerned with user-defined attributes, which is not part of the PKCS standard.

Therefore, even if a person of ordinary skill in the art were to combine the references, the alleged combination would still not arrive at the presently claimed invention. The alleged combination would be a method defined in an object type attribute values obtaining manager, as opposed to an abstract class for attribute objects. There would still be no subclass of undefined attributes or a subclass of defined attributes that comprises a subclass of PKCS defined attributes and a subclass of user-defined attributes.

In view of the above, Applicants respectfully submit that neither Katin nor RSA teach or suggest all of the features of independent claim 4. Accordingly, Applicants respectfully request the withdrawal of the rejection of claim 4 under 35 U.S.C. § 103(a). At least by virtue of their dependency on claim 4, neither Katin nor RSA teach or suggest the features of claims 5-18. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 5-18 under 35 U.S.C. § 103(a).

In addition, neither Katin nor RSA teach or suggest the specific features recited in dependent claims 5-18. For example, with regard to dependent claim 7, neither Katin nor RSA teach or suggest that each defined attribute is registered with the PKCS9 gateway class. The Office Action alleges that RSA teaches this feature on page 8, section 5.2.2, which describes an UnstructuredName attribute that includes a PKCS9String attribute type. However, the PKCS9String type is only a type that is defined to allow user to choose between IA5String type and DirectoryString in order to represent name or names of a subject as unstructured ASCII string. The PKCS9String type is not a PKCS9 gateway class, which includes PKCS-defined attributes, such as the UnstructuredName.

The Office Action alleges that it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of RSA within the system of Katin because the implementation of a different class for each

defined attribute allows the attribute object identifier to be mapped to an implementing class when instantiated. Applicants respectfully disagree. As described above, the reference of RSA describe the PKCS #9 standard that includes only PKCS standard defined attribute types, such as UnstructuredName. RSA does not mention anything about user-defined attributes. Therefore, a person of ordinary skill in the art would not have been motivated to combine the RSA with Katin, because there is no suggestion of implementing defined attributes other than PKCS #9 standard defined attributes. Thus, the alleged motivation of "implementation of different class for each defined attribute allows attribute object identifier to be mapped to an implementing class when instantiated" is not supported simply because implementation of user-defined class is not possible using the teachings of RSA.

With regard to dependent claim 9, neither Katin nor RSA teach or suggest that the user-defined attributes are registered with the PKCS9 gateway class by reading a configuration file when the PKCS9 gateway class is initially loaded. The Office Action alleges that RSA teaches these features on pages 5 and 6, in sections 4.1 and 4.2, which describes a pkcsEntity object class and a naturalPerson object class that are designed for use within directory services based on the LDAP protocol and the X.500 family of protocols. However, as described above, RSA does not teach or suggest anything about user-defined attributes. Only PKCS standard defined attributes are supported. In addition, neither the directory services based on the LDAP protocol nor the X.500 family of protocols represents a configuration file. The LDAP protocol and the X.500 family of protocols are merely standards for directory structures that may be used for organizing the attributes. There is no suggestion in RSA of implementing these protocols in a configuration file that is read when the PKCS9 gateway class is loaded. Furthermore, Katin teaches, at column 6, lines 5-32, that the object type table and the object type attribute tables are stored in a database, not a configuration file that is read when the PKCS9 gateway class is loaded. Therefore, neither Katin nor RSA teach or suggest registering user-defined attributes by reading a configuration file when the PKCS9 gateway class is loaded.

The Office Action further alleges that it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of

RSA within the system of Katin et al. because these are the general steps taken to register user-defined attributes in the PKCS9 gateway class. Applicants respectfully disagree. There is no suggestion or teaching in RSA of user-defined attributes, let alone implementing user-defined attributes in a configuration file. Katin also does not teach or suggest implementing attributes in a configuration file. Katin only teaches implementing attributes in a database table. Therefore, a person of ordinary skill in the art would not have been led to combine Katin with RSA to arrive at registering user-defined attributes by reading a configuration file when the PKCS9 gateway class is loaded simply because the alleged "general steps of registering user-defined attributes" by reading a configuration file is not supported in either reference.

With regard to dependent claim 10, neither Katin nor RSA teach or suggest a second object-oriented method that determines a type of attribute object by performing an instanceof comparison to the registered attributes. The Office Action alleges that Katin teaches these features at column 9, lines 1-17, which reads as follow:

> Similarly, the manner in which an object type attribute values obtaining manager **58a** or **58b** obtains an object type attribute entry identifier or object type attribute values, and therefore the functions provided by an object type attribute values obtaining manager **58a** or **58b**, are dependent on the object type family and the format of the object type attribute table used. For the exemplary object type attribute table format discussed earlier (**FIG. 4b**), the object type attribute values obtaining manager **58a** or **58b** may obtain the object type attribute entry identifier by matching the input arguments against the object type in the object type attribute entries and retrieving the object type attribute values from the matched object type attribute entry; and if necessary by following the chaining relationships to the chained object type attribute entries in the customized versions of the object type attribute table.

In the above section, Katin only teaches obtaining the object type entry identifier by matching an object type of the input argument against the object type in the object type attribute table. However, Katin does not teach a method that determines the type of an attribute object by performing an instanceof comparison to the registered attributes. As known by a person of ordinary skill in the art, the instanceof comparison is one of many methods for comparing objects. Katin merely teaches comparing an input argument of object type to an object type stored in the object type attribute table to determine whether they are of the same type. Katin does not specify how the object type

is compared and there is no mention of an instanceof comparison in Katin. Without the Applicants' disclosure, a person of ordinary skill in the art would not have been led to modify Katin's teaching to include an instanceof comparison. RSA also does not teach an instanceof comparison, since RSA only defines attributes to be selected for the PKCS #9 standard. RSA does not teach any implementation of the attributes. Therefore, neither Katin nor RSA teach the features of claim 10.

With regard to dependent claim 11, neither Katin nor RSA teach or suggest that the attribute object is constructed using a constructor method in a class associated with a PKCS-compatible attribute. The Office Action alleges that RSA teaches these features in the second paragraph on page 4, section 3, which reads as follows:

> Attribute types defined in this document that are useful in conjunction with storage of PKCS-related data and the pkcsEntity object class includes PKCS #12 PFX PDUs, PKCS #15 tokens and encrypted private keys.

In the above section, RSA only teaches attributes types that are useful in conjunction with storing PKCS related data and pkcsEntity object class. However, RSA does not teach how the attribute types are constructed. In the presently claimed invention, each of the PKCS defined attributes is a subclass of the defined attribute abstract class. When an attribute type is known, an attribute object may be instantiated or constructed using a constructor method of the associated PKCS defined attribute class. While RSA teaches selected attribute type classes that are used to store PKCS related data, there is no suggestion of how to instantiate these attribute type classes. Katin also does not teach how object type attributes may be constructed. Katin only teaches storing object type attributes in a database.

The Office Action alleges that it would have been obvious to a person of ordinary skill in the art to combine the teachings of RSA and Katin because these are basic steps when defining and initializing an object belonging to a class. However, as described on page 18 of the current specification, by using a constructor method in a class associated with a PKCS-compatible attribute, there is no need to convert the name value of the new attribute object to the DER encoding of its ASN.1 data type. Thus, the name of an attribute object may be directly associated with a PKCS-compatible attribute class without performing conversions when the attribute object is created. Therefore, there are

other basic steps involved when defining and initializing an object. It would not have been obvious to a person of ordinary skill in the art to avoid these basic steps without the disclosure of the Applicants. There is also no teaching or suggestion in either Katin or RSA of a constructor method in a class associated with a PKCS-compatible attribute.

With regard to dependent claim 13, neither Katin nor RSA teach or suggest in response to determining a DER-encoded byte stream with an object identifier, the second object-oriented method in the PKCS9 gateway class returns an instance of a PKCS compatible attribute. The Office Action alleges that Katin teaches these features at column 9, lines 44-56, which is reproduced above. However, in the above section, Katin merely teaches that the object type retrieval chaining key routine returns a chaining key in response to an input object type entry identifier and that the object type retrieval chained entry matching routine returns a TRUE or FALSE value in response to an input object type entry identifier and an input chaining key. Nowhere in this section, or any other section, does Katin teach or suggest that the chaining key or the object type entry identifier includes a DER-encoded byte stream with an object identifier. Katin also does not mention anything about returning an instance of PKCS compatible attribute in response to determining a DER-encoded byte stream with an object identifier. Katin merely teaches searching object type entries in an object type table of a database, not a DER-encoded byte stream, to return deriving attributes or an object type entry identifier, not an instance of PKCS compatible attribute.

RSA also does not teach returning an instance of a PKCS compatible attribute in response to determining a DER-encoded byte stream with an object identifier. While RSA defines what attributes are selected in the PKCS #9 standard, RSA does not teach returning an instance of these attributes in response to determining a DER-encoded byte stream with an object identifier.

With regard to dependent claims 14 and 15, neither Katin nor RSA teach or suggest returning an instance of an undefined attribute with a value being a DER-encoded byte stream in response to determining that the object identifier from the DER-encoded byte stream is not registered with the PKCS9 gateway class, or returning an instance of an attribute with the object identifier in response to determining that the object identifier from the DER-encoded byte stream is registered with the PKCS9 gateway class. The

Office Action alleges that Katin teaches these features at column 9, lines 38-43, which is reproduced above.

In the above section, Katin teaches an object type deriving matching routine that receives an object type entry identifier and at least one object type deriving attribute as input arguments. In response, if the deriving attribute input arguments match the object type deriving attributes in the object type entry identified by the object type entry identifier, the object type deriving matching routine returns the object type entry identifier. Otherwise, the matching routine returns a null value. Thus, Katin only determines whether an input object type deriving attribute matches the object type deriving attribute in the object type entry identified by the input object type entry identifier. Katin does not teach any determination made as to whether or not the object identifier from a DER-encoded byte stream is registered with a PKCS9 gateway class. As discussed in the arguments presented for claim 1, neither Katin nor RSA teach or suggest a PKCS9 gateway class, let alone determining whether a DER-encoded byte stream is registered with such PKCS9 gateway class.

In addition, Katin only teaches returning an object type entry identifier, which identifies an entry in the object type table that has a plurality of deriving attributes and corresponding object type, not an instance of undefined attribute with a value being a DER-encoded byte stream or an instance of attribute with object identifier, as recited in claims 14 and 15.

RSA also does not teach the features of claims 14 and 15. While RSA teaches storing attributes in a DER-encoded format, RSA does not teach a PKCS9 gateway class or determining whether or not the object identifier from the DER encoded byte stream is registered with the PKCS gateway class. Since RSA does not teach any implementation of the PKCS defined attributes, RSA would not teach the determining whether the object identifier in the PKCS defined attributes is or is not registered with the PKCS9 gateway class. Therefore, neither Katin nor RSA teach or suggest the features of claims 14 and 15.

With regard to dependent claims 16 and 17, neither Katin nor RSA teach or suggest a registered attribute object that is encoded to a DER-encoded byte stream by using the first object-oriented method for encoding the attribute object or a registered

attribute object represented as a DER-encoded byte stream that is decoded to an attribute object by using the second object-oriented method for decoding the attribute object. The Office Action alleges that RSA teaches these features on page 28, section B.3.1, which reads as follows:

> **B.3.1 userPKCS12**
> This attribute to be stored and requested in binary form, as userPKCS12;binary. The attribute values are PFX PDUs stored as binary (BER- or DER- encoded) data.
> {
> 2.16.840.1.113730.3.1.216
> NAME 'userPKCS12'
> DESC 'PKCS #12 PFX PDU for exchange of personal information'
> SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
> }

In the above section, RSA teaches a userPKCS12 attribute type that may be stored in a binary format, which is either BER or DER encoded. The above definition is in a BNF notation, which is syntax for programming language and command sets. However, RSA does not teach a method defined in an abstract class for attribute objects that encodes the userPKCS12 into DER-encoded format or a method defined in the PKCS9 gateway class that decodes a DER-encoded byte stream. RSA merely teaches storing the userPKCS12 attribute in a DER encoded binary format. RSA does not specify a method that encodes the userPKCS12 attribute or decodes the DER encoded binary stream.

The Office Action alleges that it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of RSA within the system of Katin because these are basic steps when defining and initializing an object belonging to a class. Applicants respectfully disagree. RSA only teaches the format in which an attribute object has to be stored and requested. RSA does not teach or suggest any method that encodes the attribute object or decodes the DER encoded stream. While it may be necessary for a person of ordinary skill in the art to implement a basic step to encode the attribute object to the DER encoded format prior to storing the encoded object, there is no suggestion in Katin or RSA that a method for encoding the attribute object is defined in an abstract class for attribute object or a method for decoding the DER encoded byte stream is defined in a PKCS9 gateway class, when read in combination with claim 4. Therefore, it would not have been obvious to a person of

ordinary skill in the art to include Katin and RSA's teachings to encode and decode the attribute object by using a method in an abstract attribute object class and a method in a PKCS9 gateway class, without the disclosure of Applicants.

With regard to dependent claim 18, neither Katin nor RSA teach or suggest a second object-oriented method in the PKCS9 gateway class that extracts attribute values into forms, wherein the forms are strings, numbers, and/or other non-abstract data types. The Office Action alleges that RSA teaches these features on page 29, section B.3.3, which reads as follows:

> **B.3.3 encryptedPrivateKeyInfo**
> This attribute is to be stored and requested in binary form, as encryptedPrivateKeyInfo;binary. The attribute values are EncryptedPrivateKeyInfo PDUs stored as binary (BER- or DER- encoded) data.
> {
> 1.2.840113549.1.9.25.2
> NAME 'encryptedPrivateKeyInfo'
> DESC 'PKCS #8 encrypted private key info'
> SYNTAX 1.3.6.1.4.1466.115.121.1.5
> }

In the above section, similar to reference section cited for claims 16 and 17, RSA only teaches storing and requesting attribute encryptedPrivateKeyInfo in binary format which is encoded in DER or BER. However, RSA does not teach a second object-oriented method in PKCS9 gateway class that extracts the encoded values into forms, wherein the forms include strings, numbers and/or other non abstract data types. RSA does not teach or suggest any method in any class that extracts values from encoded encryptedPrivateKeyInfo attribute into forms, let alone extracting values from the attribute into forms, such as strings, numbers, or other abstract data types.

The Office Action alleges that it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of RSA within the system of Katin because extracting values entered by the user/system into a string or etc. is a basic step in initializing an object. Applicants respectfully disagree. While it may be necessary to encode the encryptedPrivateKeyInfo attribute into DER encoding format prior to storing it, there is no teaching or suggestion that the values of the

encryptedPrivateInfo attribute are extracted into forms such as strings and number by using a method in a PKCS9 gateway class.

Neither RSA nor Katin mention anything about PKCS9 gateway class, let alone a method in PKCS9 gateway class. Katin only mentions routines that extract values from a database table, such as the object type table, but fails to mention that the routine is in a PKCS9 gateway class. RSA also fails to mention any method for extracting the encryptedPrivateKeyInfo attribute into forms. Therefore, a person of ordinary skill in the art would not have been motivated to combine the references to reach the presently claimed invention, because neither Katin nor RSA mentions anything about a method in a PKCS9 gateway class, let alone a method in a PKCS9 gateway class the extracts values into forms.

In view of the above, Applicants respectfully submits that neither Katin nor RSA teach or suggest the specific features of dependent claims 5-18. Thus, Applicant respectfully requests withdrawal of the rejection of claims 5-18 under 35 U.S.C. § 103(a).

Independent claim 22 recites:

> 22. A data processing system for managing Public Key Cryptography Standards (PKCS) compatible attributes, the data processing system comprising:
> first constructing means for constructing a new instance of an attribute object;
> first differentiating means for differentiating between attribute objects of different types;
> converting means for converting an instance of an attribute object to and/or from DER-encoding;
> first extracting means for extracting values associated with an attribute object;
> extending means for extending a set of attributes with user-defined types; and
> first registering means for registering an attribute class with a PKCS9 gateway class. (emphasis added)

Neither Katin nor RSA teach or suggest the features emphasized above. The Office Action alleges that Katin teaches constructing a new instance of an attribute object at column 2, lines 37-54, which reads as follows:

> A method and apparatus for deriving an object's object type and obtaining object type attribute values for the derived object type for an application is disclosed, which has particular application to computer systems where

applications and data manipulated by the applications are implemented in an object-oriented manner.

Under the present invention, an object type and at least one object type deriving attribute are stored as an object type entry in an object type table having an object type table identifier. The object type table is in turn stored in a database. Similarly the object type and at least one object type attribute value having a corresponding object type attribute identifier are stored as an object type attribute entry in an object type attribute table having an object type attribute table identifier. The object type attribute table is in turn also stored in the database.

In the above section, Katin teaches deriving an object type and obtaining object type attributes from an object type table and an object type attribute table stored in a database. Thus, the object type and object type attributes currently exist in the database. There is no new instance of the object type or object type attributes constructed by Katin. Therefore, Katin does not teach constructing a new instance of an attribute object, simply because the object type and object type attributes of Katin are already created and stored in the database tables.

In addition, neither Katin nor RSA teach or suggest converting an instance of an attribute object to and/or from DER-encoding. The Office Action admits that Katin does not teach DER-encoding, but the Office Action alleges that RSA teaches these features on page 28, section B.3.1, which is reproduced above in arguments presented for claim 18. As described above, RSA only teaches storing userPKCS12 attribute in a binary format, such as DER-encoded format. However, RSA does not specify any means for converting the attribute into the DER-encoded format. In the presently claimed invention, an encode method and decode method are defined in an abstract PKCSDerObject class, such that each attribute object must implement these two methods to convert an attribute object to and from DER-encoding. To the contrary, RSA does not teach such features. RSA only teaches the format in which the attribute is to be stored.

Furthermore, neither Katin nor RSA teach or suggest registering an attribute with a PKCS9 gateway class. The Office Action alleges that RSA teaches this feature on page 5, section 4.1, which is reproduced above in arguments presented for claim 1. As described above, RSA only teaches a PKCS9AttributeSet class. It is assumed that the Office Action interprets the PKCS9AttributeSet class as a PKCS9 gateway class. However, Applicants respectfully submit that the interpretation is inaccurate. The

PKCS9AttributeSet class, as defined by RSA, is an attribute that holds three other PKCS related attributes: userPKCS12, pKCS15Token and encryptedPrivateKeyInfo. These attributes relates to PKCS #12, PKCS #15 and PKCS #8 standards. Therefore, they are merely PKCS standard defined attributes. The PKCS9AttributeSet attribute is different from a PKCS9 gateway class, in that the PKCS9AttributeSet attribute does not include user-defined attributes, as described in the arguments presented for claim 1.

The Office Action alleges that it would have been obvious to combine the teachings of RSA within the system of Katin because registering an attribute in PKCS9 gateway class improves interoperability of the applications involved as explained in the claim 1 rejection. However, as described in claim 1, Katin does not teach or suggest a PKCS9 gateway class or anything related to PKCS standard. Katin only teaches retrieving and obtaining different object type and object type attributes for different applications from database tables, there is no mention of PKCS standard applications. There is also no suggestion in RSA to improve interoperability of applications by registering attributes, which includes both PKCS standard defined attributes and user-defined attributes, with a PKCS gateway class. RSA only teaches PKCS 9 standard defined attributes to be interoperable with other PKCS standards, not with user-defined attributes. Therefore, a person of ordinary skill in the art would not combine the references to reach the presently claimed invention, because there is simply no teaching or suggestion to register attributes with a PKCS9 gateway class.

In view of the above, Applicants respectfully submit that neither Katin nor RSA teach or suggest all of the features of independent claim 22. As independent claim 38 recites similar features to claim 22, claim 38 is also not taught or suggested by either reference. Accordingly, Applicants respectfully request the withdrawal of the rejection of claims 22 and 38 under 35 U.S.C. § 103(a). At least by virtue of their dependency on claims 22 and 38, respectively, neither Katin nor RSA teach or suggest the features of claims 23-34 and 39-50. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 23-34 and 39-50 under 35 U.S.C. § 103(a).

In addition, neither Katin nor RSA teach or suggest the specific features recited in dependent claims 23-34 and 39-50. For example, with regard to dependent claim 26, neither Katin nor RSA teach or suggest constructing a new instance of a PKCS-

compatible attribute object using the PKCS9 gateway class if the attribute object identifier and a class implementing that attribute are registered. The Office Action alleges that Katin teaches these features at column 10, lines 3-10, which reads as follows:

> The object type matching routine **63** receives an object type attribute entry identifier and an object type as input arguments. In response, if the object type input argument matches the object type in the object type attribute entry identified by the object type attribute entry identifier, the object type matching routine **63** returns the object type attribute entry identifier input arguments, else it returns a null value.

In the above section, Katin teaches a matching routine that takes an object type attribute entry identifier and an object type as inputs and returns the input object type attribute entry identifier if the input object type matches the object type of the entry identified by the input attribute entry identifier. Thus, Katin only teaches retrieving the object type attribute identifier that identifies an entry in the object type attribute table. Katin does not teach constructing a new instance of any object, let alone a PKCS-compatible attribute object. In addition, Katin does not teach using PKCS9 gateway class to return an instance if the attribute object identifier and a class implementing that attribute are registered. Nowhere in the above section, or any other section, does Katin teach or suggest a PKCS gateway class, let alone registering an object identifier and a class implementing that attribute with the PKCS gateway class. RSA also does not teach a PKCS gateway class. Therefore, RSA would not teach registering an object identifier and a class implementing the PKCS compatible attribute object using the PKCS9 gateway class.

With regard to claim 27, neither Katin nor RSA teach or suggest constructing a new instance of PKCS compatible attribute object using a PKCS9 gateway class based on a DER encoded byte stream. The Office Action alleges that RSA teaches these features on page 28, sections B.3.1 and B.3.3, which are reproduced above in arguments for claims 16, 17 and 18. However, in these sections, RSA merely teaches storing userPKCS12 and encryptedPrivateKeyInfo attribute in a binary form, such as DER encoding. RSA does not teach or suggest constructing a new instance of PKCS compatible attribute object based on a DER encoded byte stream. There is no mention of a PKCS9 gateway class in either reference that is used to construct the new instance. In

the presently claimed invention, a getAttribute method is provided in the PKCS9 gateway class that takes a DER encoded byte stream as its input parameter. Based on this DER encoded byte stream, the method returns a new instance of PKCS compatible attribute object constructed.

RSA does not teach or suggest such features. RSA does not construct a new instance of PKCS-compatible attribute object based on the DER encoded byte stream. Katin also does not teach these features. Katin only teaches retrieving object type and obtaining object type attribute values from the database table. Thus, the attribute values and the object type already exist in the database. Katin does not construct a new instance of the PKCS-compatible attribute object.

The Office Action alleges that it would have been obvious to one of ordinary skill in the art to combine the references, because DER encoding is a useful mean of encoding an ASN.1 value as an octet string. The Office Action further states since this works hand in hand with ASN.1, which is an OSI method of specifying abstract objects, this would be a useful way to encode the byte stream. Applicants respectfully disagree. While DER encoding is a useful way for other software to retrieve the same attribute object, there is no suggestion in either Katin or RSA of the use of a PKCS9 gateway class that constructs a new instance of PKCS-compatible attribute object based on the DER encoded byte stream. There is also no suggestion in either Katin or RSA of any specific implementation of how to construct an attribute object from a DER encoded byte stream. Therefore, a person of ordinary skill in the art with the intention to use DER to encode an attribute object would not have been led to modify the references to reach the presently claimed invention, since neither Katin nor RSA teach or suggest using a PKCS9 gateway class to construct a PKCS-compatible attribute object from the DER encoded byte stream.

In view of the above, Applicants respectfully submits that neither Katin nor RSA teach or suggest the specific features of dependent claims 23-34. Thus, Applicant respectfully requests withdrawal of the rejection of claims 23-34 under 35 U.S.C. § 103(a).

III. Conclusion

It is respectfully urged that the subject application is patentable over Katin in view of RSA and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: April 29, 2004

Stephen J. Walder, Jr.
Reg. No. 41,534
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants

SJW/im